Foundations of Informatics: a Bridging Course

Part C: Context-Free Languages
Lesson 3: Chomsky Normal Form

Thomas Noll
Software Modeling and Verification Group
RWTH Aachen University

https://moves.rwth-aachen.de/people/noll/

Software Modeling and Verification Chair
RWTH AACHEN UNIVERSITY

The Word Problem for CFL

Word Problem for CFL

Given CFG $G = \langle N, \Sigma, P, S \rangle$ and $w \in \Sigma^*$, decide whether $w \in L(G)$ or not.

The Word Problem for CFL

Word Problem for CFL

Given CFG $G = \langle N, \Sigma, P, S\rangle$ and $w \in \Sigma^*$, decide whether $w \in L(G)$ or not.

- Important problem with many applications
  - syntax analysis of programming languages
  - HTML parsers
  - ...
- For regular languages this was easy: just let the corresponding DFA run on w.
- But here: how to decide when to stop a derivation?
- **Solution:** establish normal form for grammars which guarantees that each nonterminal produces at least one terminal symbol

$\Rightarrow$ Only finitely many combinations to be inspected

Chomsky Normal Form

Definition

A CFG is in Chomsky Normal Form (Chomsky NF) if every of its productions is of the form

$$A \to BC \quad \text{or} \quad A \to a$$

Chomsky Normal Form

Definition

A CFG is in Chomsky Normal Form (Chomsky NF) if every of its productions is of the form

$$A \rightarrow BC \quad \text{or} \quad A \rightarrow a$$

Example

Consider the grammar $S \rightarrow ab \mid aSb$, which generates $L := \{a^n b^n \mid n \geq 1\}$.
An equivalent grammar in Chomsky NF is

$$\begin{aligned}
S &\rightarrow AB \mid AC && (\text{generates } L)\\
A &\rightarrow a && (\text{generates } \{a\})\\
B &\rightarrow b && (\text{generates } \{b\})\\
C &\rightarrow SB && (\text{generates } \{a^n b^{n+1} \mid n \geq 1\})
\end{aligned}$$

Conversion to Chomsky Normal Form

Theorem

Every CFL L (with $\varepsilon \notin L$) can be generated by a CFG in Chomsky NF.

Conversion to Chomsky Normal Form

Theorem

Every CFL L (with $\varepsilon \notin L$) can be generated by a CFG in Chomsky NF.

Proof.

Let L be a CFL, and let $G = \langle N, \Sigma, P, S \rangle$ be some CFG which generates L. The transformation of P into rules of the form $A \to BC$ and $A \to a$ proceeds in three steps:

1. terminal symbols only in rules of the form $A \to a$
   (thus all other rules have the shape $A \to A_1 \ldots A_n$)
2. elimination of "chain rules" of the form $A \to B$
3. elimination of rules of the form $A \to A_1 \ldots A_n$ where $n > 2$

(see following slides for details) $\square$

Step 1: Only $A \to a$

Procedure

1. For every terminal symbol $a \in \Sigma$, introduce a new nonterminal symbol $B_a \in N$.
2. Add corresponding productions $B_a \to a$ to P.
3. In each original production $A \to \alpha$, replace every $a \in \Sigma$ with B_a.

This yields G'.

Step 1: Only $A \to a$

Procedure

1. For every terminal symbol $a \in \Sigma$, introduce a new nonterminal symbol $B_a \in N$.
2. Add corresponding productions $B_a \to a$ to P.
3. In each original production $A \to \alpha$, replace every $a \in \Sigma$ with B_a.

This yields G'.

Example

$G: \; S \to ab \mid aSb$ is converted to

$$\begin{aligned} G': \; S &\to AB \mid ASB \\ A &\to a \\ B &\to b \end{aligned}$$

Step 2: Elimination of Chain Rules $A \to B$

Procedure

1. Determine all derivations $A_1 \Rightarrow \ldots \Rightarrow A_n$ with rules of the form $A \to B$ without repetition of nonterminals ($\Longrightarrow$ only finitely many!).
2. Determine all productions $A_n \to \alpha$ with $\alpha \notin N$.
3. Add corresponding productions $A_1 \to \alpha$ to P.
4. Remove all chain rules from P.

This yields G''.

Step 2: Elimination of Chain Rules $A \to B$

Procedure

1. Determine all derivations $A_1 \Rightarrow \ldots \Rightarrow A_n$ with rules of the form $A \to B$ without repetition of nonterminals ($\Longrightarrow$ only finitely many!).
2. Determine all productions $A_n \to \alpha$ with $\alpha \notin N$.
3. Add corresponding productions $A_1 \to \alpha$ to P.
4. Remove all chain rules from P.

This yields G''.

Example

$$G' : \begin{array}{lcl} S & \to & A \\ A & \to & B \mid C \\ B & \to & A \mid DA \\ C & \to & c \\ D & \to & d \end{array}$$

is converted to

$$G'' : \begin{array}{lcl} S & \to & DA \mid c \\ A & \to & DA \mid c \\ B & \to & DA \mid c \\ C & \to & c \\ D & \to & d \end{array}$$

Step 3: Elimination of Rules $A \to A_1 \ldots A_n$ with $n > 2$

Procedure

Iteratively apply the following transformation:

1. For every $A \to A_1 \ldots A_n$ with $n > 2$, introduce a new nonterminal symbol $B \in N$.
2. Replace original production by $A \to A_1 B$.
3. Add new production $B \to A_2 \ldots A_n$.

This yields G'''.

Step 3: Elimination of Rules $A \to A_1 \dots A_n$ with $n > 2$

Procedure

Iteratively apply the following transformation:

1. For every $A \to A_1 \dots A_n$ with $n > 2$, introduce a new nonterminal symbol $B \in N$.
2. Replace original production by $A \to A_1 B$.
3. Add new production $B \to A_2 \dots A_n$.

This yields G'''.

Example

$$G'': \begin{array}{rcl} S & \to & AB \mid ASB \\ A & \to & a \\ B & \to & b \end{array} \qquad \text{is converted to} \qquad G''': \begin{array}{rcl} S & \to & AB \mid AC \\ A & \to & a \\ B & \to & b \\ C & \to & SB \end{array}$$

Summary: Chomsky Normal Form

Seen:

- Chomsky NF: all productions of the form $A \rightarrow BC$ or $A \rightarrow a$

Summary: Chomsky Normal Form

Seen:

- Chomsky NF: all productions of the form $A \to BC$ or $A \to a$

Next:

- Exploit Chomsky Normal Form to solve word problem for CFL